(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication : **2 829 646**
(à n'utiliser que pour les commandes de reproduction)

(21) N° d'enregistrement national : **01 11602**

(51) Int Cl[7] : H 04 L 9/32, G 07 F 7/10, G 06 K 19/07

(12) DEMANDE DE BREVET D'INVENTION A1

(22) Date de dépôt : 07.09.01.

(30) Priorité :

(43) Date de mise à la disposition du public de la demande : 14.03.03 Bulletin 03/11.

(56) Liste des documents cités dans le rapport de recherche préliminaire : *Se reporter à la fin du présent fascicule*

(60) Références à d'autres documents nationaux apparentés :

(71) Demandeur(s) : *GEMPLUS Société anonyme* — FR.

(72) Inventeur(s) : JOYE MARC et GANDOLFI KARINE ep. VILLEGAS.

(73) Titulaire(s) :

(74) Mandataire(s) :

(54) PROCEDE SECURISE DE MISE EN OEUVRE D'UN ALGORITHME DE CRYPTOGRAPHIE ET COMPOSANT CORRESPONDANT.

(57) L'invention concerne un procédé sécurisé de mise en oeuvre, dans un composant électronique, d'un algorithme de cryptographie utilisant des moyens de calcul destinés à effectuer des opérations d'exponentiation à la puissance d d'un nombre y (y^d), d étant un nombre entier de taille déterminée. Il consiste à mémoriser plusieurs algorithmes de calcul d'exponentiation dans ledit composant électronique préalablement au premier calcul de y^d, et à réaliser les étapes suivantes à chaque calcul de la valeur y^d :

a) on décompose le nombre d en n blocs D_i de r_i bits, i variant de 0 à n-1, n et les r_i étant des entiers aléatoires, 1<n<=taille de d, $r_0+\ldots+r_{n-1}$ = taille de d et on considère parmi les algorithmes de calcul d'exponentiation mémorisés, une chaîne de n algorithmes A_i,

b) on effectue au moyen d'un algorithme A_0 et à partir des r_0 premiers bits de d et de y, le calcul

$S_0 = y^{D0}$

et on mémorise le résultat S_0,

c) on effectue au moyen d'un algorithme A_j et à partir des r_j bits suivants de d, d'une fonction de S_0 et/ou... et/ou de S_{j-1}, et de y, j variant de 1 à n-1, le calcul

$S_j = y^{Dj=fDj\ldots}$

et on mémorise le résultat S_j,

le résultat de y^d correspondant à la valeur S_{n-1} obtenue.

FR 2 829 646 - A1

PROCÉDÉ SÉCURISÉ DE MISE EN ŒUVRE D'UN ALGORITHME DE CRYPTOGRAPHIE ET COMPOSANT CORRESPONDANT

La présente invention concerne un procédé sécurisé de mise en œuvre, dans un composant électronique, d'un algorithme de cryptographie utilisant le calcul d'exponentiation à la puissance d d'un nombre y.

L'invention se rapporte également au composant électronique correspondant.

De tels composants sont utilisés dans des applications où l'accès à des services ou à des données est sévèrement contrôlé. Ils ont une architecture formée autour d'un microprocesseur et de mémoires, dont une mémoire programme de type ROM ("Read Only Memory" en anglais) qui contient le(s) nombre(s) secret(s) d.

Ces composants sont utilisés dans des systèmes informatiques, embarqués ou non ; ils sont notamment utilisés dans les cartes à puce, pour certaines applications de celles-ci. Ce sont par exemple des applications d'accès à certaines banques de données, des applications bancaires, des applications de télépéage, par exemple pour la télévision, la distribution d'essence ou encore le passage de péages d'autoroutes.

Ces composants ou ces cartes mettent donc en œuvre un algorithme de cryptographie pour assurer le chiffrement de données émises et/ou le déchiffrement de données reçues lorsque celles-ci doivent demeurer confidentielles.

De manière générale et succincte, ces algorithmes cryptographiques ont notamment pour fonction le chiffrement ou la signature numérique d'un message

appliqué en entrée (à la carte) par un système hôte (serveur, distributeur bancaire...) et de nombreux secrets contenus dans la carte, et de fournir en retour au système hôte ce message chiffré ou signé, ce qui permet par exemple au système hôte d'authentifier le composant ou la carte, d'échanger des données, ….

Les caractéristiques des algorithmes de cryptographie sont connues : calculs effectués, paramètres utilisés. La seule inconnue est le ou les nombres secrets contenus en mémoire programme. Toute la sécurité de ces algorithmes de cryptographie tient dans ce(s) nombre(s) secret(s) contenu(s) dans la carte et inconnu(s) du monde extérieur à cette carte. Ce nombre secret ne peut être déduit de la seule connaissance du message appliqué en entrée et du message chiffré fourni en retour.

Or il est apparu que des attaques externes basées sur des grandeurs physiques mesurables à l'extérieur du dispositif de calcul lorsque celui-ci est en train de dérouler l'algorithme de cryptographie, permettent à des tiers mal intentionnés de trouver le(s) nombre(s) secret(s) contenu(s) dans cette carte. Ces attaques sont appelées attaques à canaux cachés ("Side channel attacks" en anglais) ; on distingue parmi ces attaques à canaux cachés, les attaques SPA acronyme anglo-saxon pour *Single Power Attack* et les attaques DPA, acronyme anglo-saxon pour *Differential Power Analysis*.

Le principe de ces attaques à canaux cachés repose par exemple sur le fait que la consommation en courant du microprocesseur exécutant des instructions varie selon l'instruction ou la donnée manipulée.

Notamment, quand une instruction exécutée par le microprocesseur nécessite une manipulation d'une donnée ou d'une instruction bit par bit, on a deux profils de courant différents selon que ce bit vaut "1" ou "0". Typiquement, si le microprocesseur manipule un "0", on a à cet instant d'exécution une première amplitude du courant consommé et si le microprocesseur manipule un "1", on a une deuxième amplitude du courant consommé, différente de la première.

Ainsi les attaques à canaux cachés exploitent dans ce cas la différence du profil de consommation en courant dans la carte pendant l'exécution d'une instruction suivant la valeur du bit manipulé. Pour une description plus détaillée des attaques à canaux cachés, on se reportera à la publication de Paul Kocher "Advances in Cryptology-CRYPTO'99", vol. 1666 of Lectures Notes in Computer Science, pp 388-397, Springer Verlag 1999.

Ce type d'attaque est notamment envisageable avec l'algorithme RSA du nom de ses auteurs (Rivest, Shamir et Adleman), sur lequel sont basés de nombreux algorithmes de cryptographie. La sécurité de l'algorithme RSA est basée sur la difficulté de factoriser de grands nombres. Ces algorithmes utilisent notamment des calculs d'exponentiation à la puissance d, d étant un nombre secret.

On rappelle brièvement les principales étapes de l'algorithme RSA.

On établit un nombre N qui est le produit de deux nombres premiers p et q (N=p.q), ainsi qu'un exposant

public ou clé publique e et un exposant privé ou clé privée ou secrète d, satisfaisant la relation :

$$e.d = 1 \text{ (modulo } \lambda(N)),$$

$\lambda(.)$ étant la fonction de Carmichael.

Selon un premier mode de fonctionnement de l'algorithme RSA dit classique, les paramètres publics sont (N,e) et les paramètres privés sont (N,d). Etant donné x compris dans l'intervalle]0,N[, l'opération publique sur x qui peut être par exemple le chiffrement du message x ou encore la vérification de la signature x, consiste à calculer :

$$y = x^e \text{ modulo } N$$

L'opération privée correspondante qui peut être par exemple le déchiffrement du message chiffré y ou la génération d'une signature x, est alors :

$$x = y^d \text{ modulo } N$$

Un autre mode de fonctionnement dit mode CRT car basé sur le théorème des restes chinois ("Chinese Remainder Theorem" ou CRT en anglais) est quatre fois plus rapide que celui de l'algorithme RSA classique. Selon ce RSA mode CRT, on n'effectue pas directement les calculs modulo N mais on effectue d'abord les calculs modulo p et modulo q.

Les paramètres publics sont (N,e) mais les paramètres privés sont (p,q,d) ou (p,q,d_p,d_q,i_q) avec

$$d_p = d \text{ modulo}(p-1),\ d_q = d \text{ modulo}(q-1)$$

$$\text{et} \quad i_q = q^{-1} \text{ modulo } p.$$

L'opération publique s'effectue de la même façon que pour le mode de fonctionnement classique ; par contre pour l'opération privée, on calcule d'abord :

$$x_p = y^{d_p} \text{mod} p \qquad \text{et} \qquad x_q = y^{d_q} \text{mod} q$$

Ensuite, par application du théorème des restes chinois, on obtient $x = y^d \bmod N$ par :

$$x = x_q + q[i_q(x_p - x_q) \text{modulo } p]$$

Ainsi en mesurant la consommation en courant de la carte par exemple, lors de ces calculs d'exponentiation à la puissance d, un fraudeur peut déduire la suite de bits composant la clé d à partir de multiples mesures.

Les procédés de contre-mesure permettant de parer ce type d'attaque consistent à ne pas manipuler directement la clé secrète d.

Selon une première contre-mesure, on décompose d sous la forme suivante d =(d+a)-a et l'on calcule alors $y^{(d+a)}$ puis y^{-a}. Mais a et d étant de même ordre de grandeur, le temps de calcul est alors doublé.

Une autre contre-mesure seulement utilisable en mode CRT et décrite dans le brevet WO n°9935782 consiste à exprimer l'exposant privé d_p sous la forme aléatoire suivante :

$d_p^* = d_p + r.(p-1)$, r étant un nombre aléatoire.

Mais cette contre-mesure ne peut s'appliquer au mode RSA classique.

Une troisième contre-mesure consiste à représenter l'exposant d par une chaîne d'addition. Par exemple, une chaîne d'addition possible du nombre 10 est (1,2,3,5,8,10). Pour calculer y^{10}, on calcule $R1=y.y=y^2$ puis $R2=R1.y=y^2.y=y^3$ puis $R3=R2.R1=y^3.y^2=y^5$ puis $R4=R3.R2=y^5.y^3$ et enfin $R5=R4.R1=y^8.y^2=y^{10}$. Bien sûr dans la réalité la clé secrète d est un nombre beaucoup plus grand, de 512 bits ou plus.

Pour une description plus détaillée des représentations d'un nombre par chaînes d'addition, on peut se reporter à la publication de C. Clavier et de M. Joye CHESS'01 "Universal exponentiation algorithm : a first step toward provable SPA resistance".

Dans ce cas de contre-mesure, la taille des données à garder en mémoire en l'occurrence la chaîne d'addition, est doublée.

Le but de la présente invention est donc de protéger le nombre secret d contre les attaques à canaux cachés intervenant lorsque d est utilisé dans des calculs d'exponentiation, sans pour autant être pénalisé au niveau du temps de calcul, de l'emplacement mémoire ou encore limité dans le choix de l'algorithme de cryptographie.

L'invention a pour objet un procédé sécurisé de mise en œuvre, dans un composant électronique, d'un algorithme de cryptographie utilisant des moyens de calcul destinés à effectuer des opérations d'exponentiation à la puissance d d'un nombre y (y^d), d étant un nombre entier de taille déterminée, caractérisé en ce qu'on mémorise plusieurs algorithmes de calcul d'exponentiation dans ledit composant électronique préalablement au premier calcul de la valeur y^d, et en ce qu'il consiste à réaliser les étapes suivantes à chaque calcul de la valeur y^d :

a) on décompose le nombre d en n blocs D_i de r_i bits, i variant de 0 à n-1, n et les r_i étant des entiers aléatoires, $1<n<=$taille de d, $r_0+\ldots+r_{n-1}=$taille de d et on considère parmi les algorithmes de calcul

d'exponentiation mémorisés, une chaîne de n algorithmes A_i,

b) on effectue au moyen d'un algorithme A_0 et à partir des r_0 premiers bits de d et de y, le calcul

$$S_0 = y^{D_0}$$

et on mémorise le résultat S_0,

c) on effectue au moyen d'un algorithme A_j et à partir des r_j bits suivants de d, d'une fonction de S_0 et/ou ... et/ou de S_{j-1}, et de y, j variant de 1 à n-1, le calcul

$$S_j = y^{D_0 \| \ldots \| D_j}$$

et on mémorise le résultat S_j,

le résultat de y^d correspondant à la valeur S_{n-1} obtenue.

Selon une caractéristique de l'invention, parmi la chaîne des n algorithmes A_i, certains desdits algorithmes sont identiques.

La chaîne des n algorithmes A_i i variant de 0 à n-1, est de préférence établie de manière aléatoire.

La chaîne des n algorithmes A_i i variant de 0 à n-1, peut être prédéterminée.

Selon un mode de réalisation de l'invention, l'algorithme de cryptographie est du type RSA classique ou en mode CRT.

L'invention a également pour objet un composant électronique de sécurité, comprenant des moyens de calcul, une mémoire de programme et une mémoire de travail et des moyens de communication de données, caractérisé en ce qu'il met en œuvre le procédé de

contre-mesure selon l'une quelconque des revendications précédentes, et en ce qu'il comprend un générateur de nombres aléatoires et en ce que la mémoire de programme comporte plusieurs algorithmes de calcul d'exponentiation.

L'invention concerne aussi une carte à puce comprenant un composant électronique tel que décrit précédemment.

D'autres particularités et avantages de l'invention apparaîtront clairement à la lecture de la description faite à titre d'exemple non limitatif et en regard de la figure 1 annexée qui représente schématiquement les éléments d'une carte à puce apte à mettre en œuvre l'invention.

Les modes de réalisation sont décrits dans le cadre de cartes à puce, mais peuvent bien entendu s'appliquer à tout autre dispositif ou composant électronique de sécurité doté de moyens de calculs cryptographiques.

Ainsi que le montre la figure 1, la carte à puce 1 comprend un microprocesseur 2 couplé à une mémoire figée (ROM) 3 et à une mémoire vive (RAM) 4, le tout formant un ensemble permettant, entre autres, l'exécution d'algorithmes cryptographiques. Plus précisément, le microprocesseur 2 comporte les moyens de calcul arithmétiques nécessaires à l'algorithme, ainsi que des circuits de transfert de données avec les mémoires 3 et 4. La mémoire figée 3 contient le programme exécutoire de l'algorithme cryptographique sous forme de code source, alors que la mémoire vive 4

comporte des registres pouvant être mis à jour pour le stockage de résultats des calculs.

La carte à puce 1 comporte aussi une interface de communication 5 reliée au microprocesseur 2 pour permettre l'échange de données avec l'environnement extérieur. L'interface de communication 5 peut être du type "à contacts", étant dans ce cas formée d'un ensemble de plots de contacts destinés à se connecter à un contacteur d'un dispositif externe, tel qu'un lecteur de cartes, et/ou du type "sans contact". Dans ce dernier cas, l'interface de communication 5 comporte une antenne et des circuits de communication par voie hertzienne permettant un transfert de données par liaison sans fil. Cette liaison peut aussi permettre un transfert d'énergie d'alimentation des circuits de la carte 1.

Le procédé selon l'invention consiste à calculer y^d au moyen d'une chaîne de n algorithmes d'exponentiation appliqués à n blocs de bits de d. On entend par algorithme de calcul d'exponentiation, une suite d'instructions permettant d'effectuer ce calcul ; comme on le verra par la suite dans un exemple, il y a plusieurs algorithmes possibles.

Cette chaîne de n algorithmes d'exponentiation A_i (i variant de 0 à n-1) est établie à partir de plusieurs algorithmes d'exponentiation que l'on met en mémoire préalablement au premier calcul de y^d. Les algorithmes A_i ne sont pas nécessairement tous différents : certains peuvent être identiques.

A chaque calcul de la valeur y^d, par exemple à chaque fois que l'on effectue l'opération privée

décrite précédemment pour l'algorithme RSA, on découpe comme représenté ci-dessous la clé d comportant K bits (la taille de d est K), en n blocs D_i (i varie de 0 à n-1) de r_i bits (0 ou 1), n et r_i étant des nombres entiers aléatoires, 1<n<=K générés par exemple par la carte à puce au moyen d'un générateur 6 de nombres aléatoires représenté figure 1 :

$$d = \underbrace{d_k \ldots\ldots\ldots\ldots d_j \ldots\ldots\ldots d_0}_{\substack{r_0 \quad \ldots \quad r_i \quad \ldots \quad r_{n-1} \\ D_0 \quad \ldots \quad D_i \quad \ldots \quad D_{n-1}}}$$

(représentation binaire de d)

(nombre de bits de chaque bloc D)

avec $D_0=r_0$ premiers bits de d,

$D_1=r_1$ bits suivant ceux du bloc D_0,

...,

$D_i= r_i$ bits suivant ceux des blocs $D_0||..||D_{i-1}$,

...,

$D_{n-1}= r_{n-1}$ derniers bits de d,

avec $d= D_0||..||D_{n-1}$, le symbole || signifiant concaténé.

Les blocs D_i qui se suivent sont contigus.

La clé d est lue de gauche à droite ; on peut bien sûr tout aussi bien la lire de droite à gauche. De même on peut lire les bits d'un bloc de gauche à droite et ceux d'un autre bloc de droite à gauche.

On effectue ensuite le calcul d'exponentiation y^d au moyen de la chaîne de n algorithmes :

- on calcule au moyen d'un algorithme A_0 le calcul d'exponentiation de y à partir des r_0 premiers bits de d et de y pour obtenir S_0 que l'on mémorise :

$$S_0 = y^{D_0}$$

- on calcule au moyen d'un algorithme A_1 le calcul d'exponentiation de y à partir des r_1 bits suivants de d, d'une fonction de S_0 et de y pour obtenir S_1 que l'on mémorise :

$$S_1 = y^{D_0|D_1}$$

...,

- on calcule au moyen d'un algorithme A_i le calcul d'exponentiation de y à partir des r_i bits suivants de d, d'une fonction de S_0 et/ou de S_1 et/ou ...,et/ou de S_{i-1}, et de y pour obtenir S_i que l'on mémorise :

$$S_i = y^{D_0|D_1|\cdots|D_i}$$

...,

- on calcule au moyen d'un algorithme A_{n-1} le calcul d'exponentiation de y à partir des r_{n-1} derniers bits de d, d'une fonction de S_0 et/ou de S_1 et/ou ..., et/ou de S_{n-2}, et de y pour obtenir S_{n-1}, le résultat recherché, que l'on mémorise :

$$S_{n-1} = y^{D_0|\cdots|D_{n-1}} = y^d$$

Selon l'algorithme A_i mis en œuvre, le calcul de S_i s'effectue à partir notamment d'une fonction du résultat S_{i-1} de l'algorithme précédent sans utiliser toute la chaîne S_0, ..., S_{i-2} des résultats précédents (ou d'une fonction des résultats précédents), ou encore à

partir de quelques uns de ces résultats (ou d'une fonction de ces résultats).

Par ailleurs, la chaîne d'algorithme est de préférence déterminée de manière aléatoire pour chaque calcul de y^d ; elle peut cependant être prédéterminée.

Ce procédé permet ainsi de se protéger dans le cas de la mise en œuvre d'un algorithme de cryptographie basé sur un calcul d'exponentiation du type y^d.

En effet, les attaques à canaux cachés qui sont réalisées au terme de multiples mesures, consistent dans un premier temps, lors d'une phase d'apprentissage à identifier l'algorithme d'exponentiation utilisé. Pour ce faire, le fraudeur relance l'algorithme en changeant la clé. Connaissant cet algorithme, il peut alors identifier les bits de la véritable clé d.

Selon l'invention, le calcul de y^d n'est pas effectué de la même manière d'une fois sur l'autre : la décomposition de d en blocs D_i est déterminée de façon aléatoire d'une fois sur l'autre car le nombre n de blocs et leur taille r_i sont des entiers aléatoires, les algorithmes de calcul d'exponentiation diffèrent les uns des autres (même si certains sont identiques, il y en a au moins deux qui diffèrent) et éventuellement, la chaîne d'algorithmes d'exponentiation varie également de façon aléatoire d'un calcul de y^d à l'autre.

Il devient ainsi beaucoup plus difficile voire impossible d'identifier l'algorithme d'exponentiation utilisé et par conséquent de déterminer l'exposant secret.

On va à présent illustrer le procédé précédemment décrit par l'exemple suivant qui consiste à réaliser le

calcul d'exponentiation modulaire suivant : y^d modulo p, d étant une clé de k bits et p étant un nombre premier.

On décompose la clé d en 2 blocs (n=2), un bloc D_0 de k-r bits et un bloc D_1 de r bits ($d=D_0||D_1$).

Le calcul "y^d modulo p" va donc s'effectuer au moyen de deux algorithmes A_0 et A_1 permettant d'obtenir respectivement S_0 et S_1 avec $S_1=y^d$ modulo p.

L'algorithme choisi pour A_0 est l'algorithme "Square and Always Multiply" qui consiste à élever au carré et à multiplier dans tous les cas. Il s'agit, pour des exposants de même taille, d'un algorithme à temps constant et à code constant c'est-à-dire qui exécute toujours les mêmes instructions, quelle que soit la valeur de l'exposant manipulé ; il n'inclut pas d'instructions de test pouvant donner une information sur la valeur manipulée. Un tel algorithme à temps constant et à code constant est bien sécurisé mais il est coûteux en temps de calcul car il exécute parfois des opérations inutiles.

L'algorithme choisi pour A_1 est l'algorithme "Square and Multiply" qui consiste à élever au carré et à multiplier. Il ne s'agit pas d'un algorithme à temps constant, ni à code constant sauf si l'on rajoute artificiellement une instruction comme on le verra plus loin. C'est donc un algorithme moins sécurisé que le précédent mais plus rapide.

On calcule donc d'abord S_0 au moyen de A_0 à partir de 1, y, des k-r premiers bits de d et de p selon la suite d'instructions suivante :

R0=1

Pour i=k-1 à r par pas de (-1)

R1=$R0^2$ modulo p

R0=R1*y

R0=R[complément de d_i]

FinPour

S_0=R0

On rappelle que si d_i=0, le complément de d_i est 1 et R[complément de d_i]=R1, et que si d_i=1, le complément de d_i est 0 et R[complément de d_i]=R0.

On enchaîne ensuite avec le calcul de S_1 (c'est-à-dire y^d modulo p) au moyen de A_1 à partir de S_0, y, des r bits suivants de d et de p selon la suite d'instructions suivante :

R0=S_0

Pour i=r-1 à 0 par pas de (-1)

R1=$R0^2$

si d_i=1 alors R0=R1*y modulo p

sinon R0=R1

FinPour

S_1=R0

Pour que l'algorithme A1 soit à code constant, il suffit de remplacer l'instruction R0=R1 par :

R0=R1*1 modulo p.

L'invention est valable pour les algorithmes cryptographiques utilisant des calculs d'exponentiation comme par exemple le calcul d'exponentiation modulaire (y^d modulo p) dans le cas de l'algorithme de type RSA ou encore le calcul d'exponentiation sur une courbe elliptique où il est d'usage d'écrire l'exponentiation

de façon additive. L'exponentiation se nomme alors multiplication scalaire :

$y^b=y*\ldots*y$ (b fois) devient $b*y=y+\ldots+y$ (b fois).

REVENDICATIONS

1. Procédé sécurisé de mise en œuvre, dans un composant électronique, d'un algorithme de cryptographie utilisant des moyens de calcul destinés à effectuer des opérations d'exponentiation à la puissance d d'un nombre y (y^d), d étant un nombre entier de taille déterminée, caractérisé en ce qu'on mémorise plusieurs algorithmes de calcul d'exponentiation dans ledit composant électronique préalablement au premier calcul de la valeur y^d, et en ce qu'il consiste à réaliser les étapes suivantes à chaque calcul de la valeur y^d :

a) on décompose le nombre d en n blocs D_i de r_i bits, i variant de 0 à n-1, n et les r_i étant des entiers aléatoires, 1<n<=taille de d, $r_0+\ldots+r_{n-1}$=taille de d et on considère parmi les algorithmes de calcul d'exponentiation mémorisés, une chaîne de n algorithmes A_i,

b) on effectue au moyen d'un algorithme A_0 et à partir des r_0 premiers bits de d et de y, le calcul

$$S_0 = y^{D_0}$$

et on mémorise le résultat S_0,

c) on effectue au moyen d'un algorithme A_j et à partir des r_j bits suivants de d, d'une fonction de S_0 et/ou … et/ou de S_{j-1}, et de y, j variant de 1 à n-1, le calcul

$$S_j = y^{p_0 \| \ldots \| p_j}$$

et on mémorise le résultat S_j,
le résultat de y^d correspondant à la valeur S_{n-1} obtenue.

2. Procédé selon la revendication précédente, caractérisé en ce que parmi la chaîne des n algorithmes A_i, certains desdits algorithmes sont identiques.

3. Procédé selon l'une quelconque des revendications précédentes, caractérisé en ce que la chaîne des n algorithmes A_i i variant de 0 à n-1, est établie de manière aléatoire.

4. Procédé selon l'une quelconque des revendications 1 à 2, caractérisé en ce que la chaîne des n algorithmes A_i i variant de 0 à n-1, est prédéterminée.

5. Procédé selon l'une quelconque des revendications précédentes, caractérisé en ce que l'algorithme de cryptographie est du type RSA.

6. Procédé selon la revendication précédente, caractérisé en ce que l'algorithme de type RSA est du type CRT.

7. Composant électronique de sécurité, comprenant des moyens de calcul (2), une mémoire de programme (3) et une mémoire de travail (4) et des moyens de

communication de données (5), caractérisé en ce qu'il met en œuvre le procédé de contre-mesure selon l'une quelconque des revendications précédentes, et en ce qu'il comprend un générateur (6) de nombres aléatoires et en ce que la mémoire de programme (3) comporte plusieurs algorithmes de calcul d'exponentiation.

8. Carte à puce comprenant un composant électronique selon la revendication précédente.

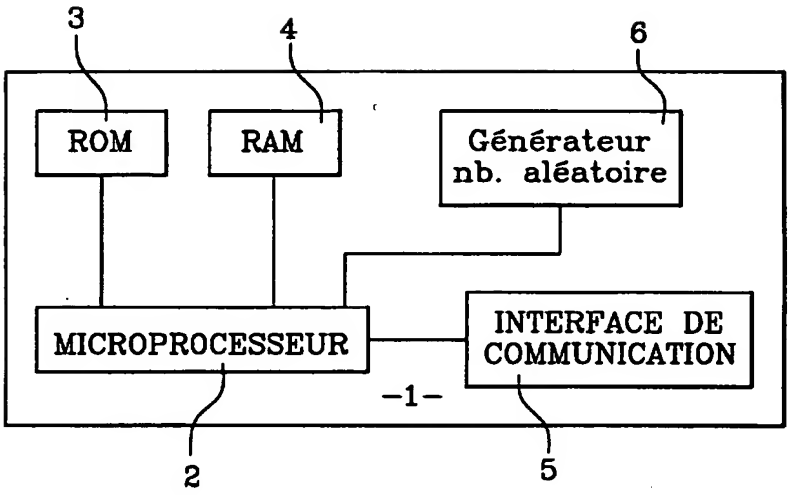
3
4
6
ROM
RAM
Générateur
nb. aléatoire
MICROPROCESSEUR
INTERFACE DE
COMMUNICATION
-1-
2
5

Fig. 1

RÉPUBLIQUE FRANÇAISE

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

2829646

N° d'enregistrement national

FA 606793
FR 0111602

RAPPORT DE RECHERCHE PRÉLIMINAIRE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	WO 00 67410 A (MOTOROLA INC) 9 novembre 2000 (2000-11-09) * abrégé; figure 3 *	1-8	H04L9/32 G07F7/10 G06K19/07
A	US 6 064 740 A (CURIGER ANDREAS ET AL) 16 mai 2000 (2000-05-16) * figures 3,4 *	1-8	
A	MENEZES ALFRED J ET AL: "Handbook of Cryptography" HANDBOOK OF APPLIED CRYPTOGRAPHY, CRC PRESS SERIES ON DISCRETE MATHEMATICES AND ITS APPLICATIONS, BOCA RATON, FL, CRC PRESS, US, 1997, pages 613-629, XP002188328 ISBN: 0-8493-8523-7 * page 614 - page 616; tableau 14.15 *	1-8	

DOMAINES TECHNIQUES RECHERCHÉS (Int.Cl.7)

G06F

Date d'achèvement de la recherche: 19 avril 2002

Examinateur: Verhoof, P

CATÉGORIE DES DOCUMENTS CITÉS

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : arrière-plan technologique
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

EPO FORM 1503 12.99 (P04C14)

1

2829646

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 0111602 FA 606793

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.
Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du 19-04-2002
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
WO 0067410 A	09-11-2000	US 6298135 B1	02-10-2001
		AU 4673900 A	17-11-2000
		WO 0067410 A1	09-11-2000
US 6064740 A	16-05-2000	AUCUN	

EPO FORM P0465

Pour tout renseignement concernant cette annexe : voir Journal Officiel de l'Office européen des brevets, No.12/82